

VISITA WWW.RASPBERRYITALY.COM

The MagPi

Numero 82 | Giugno 2019 | magpi.cc raspberryitaly.com

La rivista ufficiale Raspberry Pi tradotta in italiano da RaspberryItaly

IMPARARE A PROGRAMMARE

CON SCRATCH & PYTHON

INOLTRE:

- Progetti di modellismo ferroviario
- Le migliori risorse per imparare l'intelligenza artificiale

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. ISSN: 2051-9982.

IMPARARE A PROGRAMMARE

CON SCRATCH E PYTHON

Chiunque può imparare a programmare! Programmare un computer è più facile di quanto pensi. Seguici e ti aiuteremo a partire

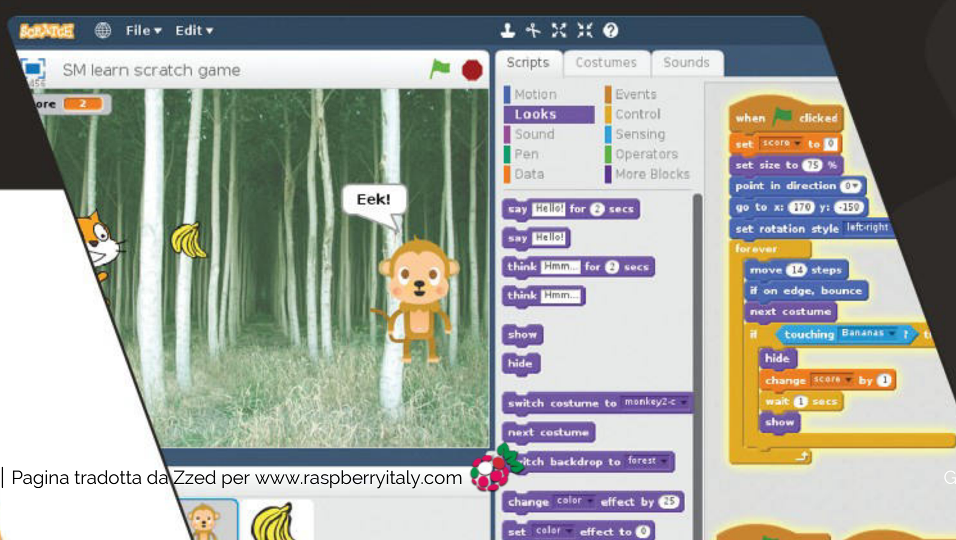


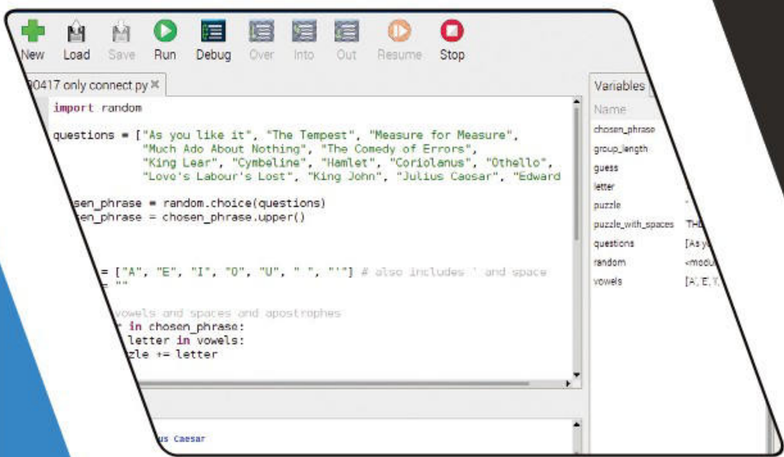
AUTORE

Sean McManus

Autore/co-autore degli ispiranti libri di programmazione Incluso Mission Python, Cool Scratch Projects in Easy Step e Raspberry Pi for Dummies. Ottieni dei capitoli gratuiti sul sito Web di Sean.

sean.co.uk





Imparare a programmare può essere una delle abilità più profonde che mai svilupperai. Con il codice, puoi controllare un computer. Puoi fargli fare delle cose per te, o anche controllare dei gadget e altri computer. Rilassati e lascia che il computer faccia tutto il lavoro.

Certo, è bello. Ma il codice è più di questo. Si tratta di capire come funzionano i computer e di avere una migliore comprensione di come funziona la tecnologia - e il mondo moderno. Si tratta di suddividere i problemi in pochi bit e risolverli. È un'abilità incredibilmente utile. Ecco perché è profonda.

A un livello più pratico, sapere un po' di programmazione può portare a migliori opportunità di lavoro; può aprire un po' le porte a lavori ben pagati e divertenti. È un'abilità impressionante da mettere nel tuo CV e chiunque può farlo. Chiunque.

La programmazione è molto più semplice di quanto pensi. E mettere la potenza dell'informatica e del making digitale nelle mani delle persone è quello per cui esiste Raspberry Pi.

Raspberry Pi è "il piccolo computer che potrebbe", e tu sei "la persona che può". Non preoccuparti: sei nel posto giusto. Possiamo aiutarti ad iniziare.

> Raspberry Pi è 'il piccolo computer che potrebbe' **e tu sei 'la persona che può'**

Comincia a programmare con Scratch

Principianti, alzatevi! È tempo di fare i primi passi con la programmazione, quindi ti introduciamo a Scratch e Python, con una spruzzata di LED scintillanti! Di **Sean McManus**

Essere in grado di scrivere programmi è come avere un superpotere: significa che puoi far fare al computer tutto quello che vuoi. Unisciti a noi e ti mostreremo come realizzare i primi programmi usando Scratch e Python. Vedrai anche quanto è facile costruire semplici progetti di elettronica.

Un programma è solo un insieme di istruzioni. In Scratch, le istruzioni sono scritte con blocchi visuali che si uniscono insieme per creare una sequenza chiamata script.

I blocchi sono codificati per colore per aiutarti a trovarli. Per trovare i blocchi marroni, ad esempio, fai clic sul pulsante marrone Events sopra la Tavolozza dei blocchi.

Scratch rende la codifica più facile, perché non devi preoccuparti della sintassi dei comandi. E tutto è organizzato di fronte a te.

Comprendere le coordinate

La schermata di Scratch è suddivisa in unità chiamate passi. Quando il gatto si muove di 10 passi, fa solo un movimento, ma quella mossa lo sposta di 10 posizioni attraverso l'area. Il centro dell'area è x = 0, y = 0. L'asse x (da sinistra a destra) va da -240 a +240 e l'asse y (dal basso verso l'alto) va da -180 a +180. Le direzioni di movimento dello sprite sono numerate 0 (su), 90 (a destra), 180 o -180 (in basso) e -90 (a sinistra). Puoi usare anche numeri intermedi tra questi, quindi -45 sarebbe in direzione Nord-Ovest. Perché non provare a iniziare un nuovo progetto unendo alcuni blocchi di movimento tra loro per sperimentare? È possibile eseguire uno script o un blocco facendo clic su di esso, oppure usando il blocco when bandiera clicked come abbiamo fatto in questo programma.

Cosa Serve

- Scratch 2
- Raspbian con Desktop e il Software Raccomandato
- Un buon senso del ritmo!

01 Avvia Scratch 2

Apri Scratch facendo clic sull'icona menù di Raspberry Pi e scegliendo Programmazione> Scratch 2. Vedrai l'interfaccia di Scratch e un personaggio singolo (un gatto) in alto a sinistra, noto come "Scratch Cat".

Per controllare lo Scratch Cat, trascineremo blocchi dalla tavolozza nell'area degli script e li uniremo tra loro.

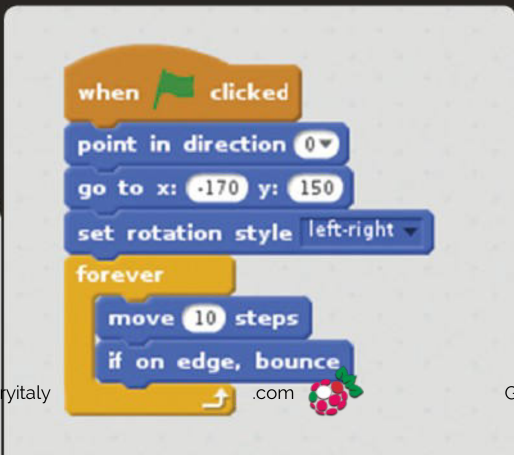
Inizia facendo clic su Eventi e trascina il blocco when clicked nell'area degli script.

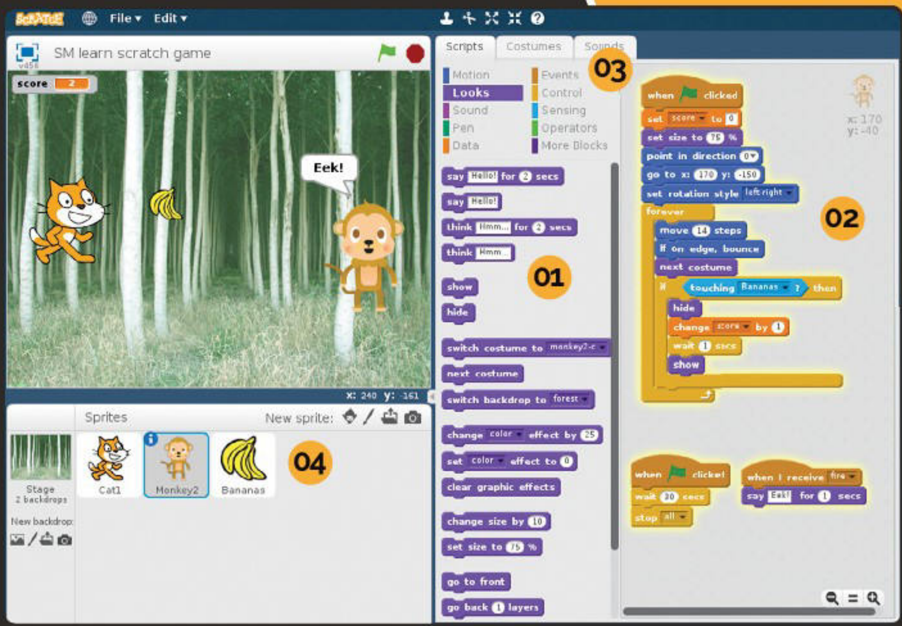
Ora fai clic su Motion e trascina il blocco point in direction 90 e connettilo alla parte inferiore del blocco when clicked .

Clicca sui campi nei blocchi per modificare il numero. Clicca su '90' e cambialo in '0'.

Ora fai clic e trascina i blocchi sottostanti e modifica i loro numeri, per costruire uno script per Scratch Cat. Questo script viene eseguito quando clicchi sulla bandiera verde sopra il palco. Imposta la direzione di movimento del gatto verso l'alto, metti il gatto nell'angolo in alto a sinistra del palco e impostalo in modo che sia sempre rivolto a sinistra o a destra. Quindi, il blocco movimento all'interno della parentesi forever mantiene il gatto in movimento tutto il tempo.

Fai clic sulla bandiera verde per eseguire il tuo script. Lo Scratch Cat si sposterà sul lato sinistro e rimbalzerà su e giù.





01 **Blocks Palette:** Qui trovi i blocchi istruzioni

02 **Scripts Area:** Trascina qui i blocchi costruire il tuo script (programma)

03 **Buttons:** Clicca per vedere diversi tipi di blocchi nella Blocks Palette sottostante

04 **Sprite List:** qui scegli e gestisci i tuoi sprite

02 Invio di una trasmissione

Gli oggetti in movimento in Scratch, gatto incluso, sono chiamati sprite. Uno sprite può inviare un messaggio a tutti gli altri sprite usando un broadcast. Non puoi sentirlo o vederlo sullo schermo, ma gli sprite possono ascoltarlo e poi iniziare uno script quando lo hanno ricevuto. Useremo un broadcast per fare lanciare delle banane al gatto. Fai clic sul pulsante marrone degli eventi e aggiungi i due blocchi nell'area degli script. Questo nuovo script non si unisce all'esistente (dal passo 1) – è nella stessa Area degli script, ma indipendente.

Devi cambiare "message1" in "fire". Clicca sulla freccia giù accanto a "messagge1" nel blocco broadcast e scegli New Message. Inserisci il messaggio chiamato "fire" e fai clic su OK.

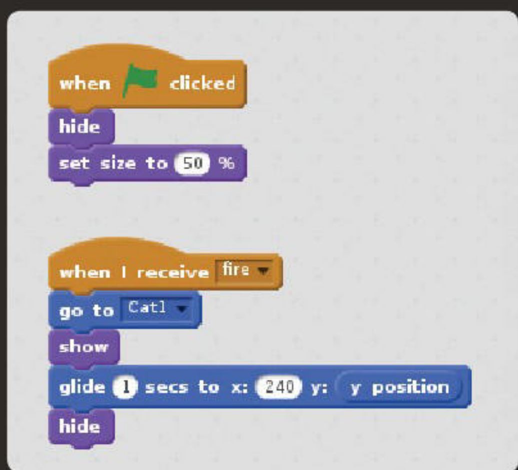
Mentre il programma è in esecuzione, quando si preme SPAZIO, il messaggio broadcast viene inviato silenziosamente. Furbo!



03 Aggiungi banane aerodinamiche

Scratch ti consente di ottenere risultati velocemente perché include proprie immagini e suoni. Fai clic sul primo pulsante New Sprite nella Sprite List (verrà visualizzato 'Scegli sprite dalla libreria' come ci si posiziona sopra di esso).

Fai clic sullo sprite Bananas per aggiungerlo alla Sprite List. Si noti che l'area degli script è vuota. Bene, dai a Bananas due script. Il primo imposta la dimensione dello sprite e lo rende invisibile quando si fa clic sulla bandiera verde. Il secondo viene lanciato quando il gatto trasmette il suo messaggio fire. Clicca e trascina i blocchi sotto nell'area degli script.



Quando aggiungi il blocco glide, trascina il blocco di y position all'interno del campo'y:' per sostituire il valore predefinito "10".

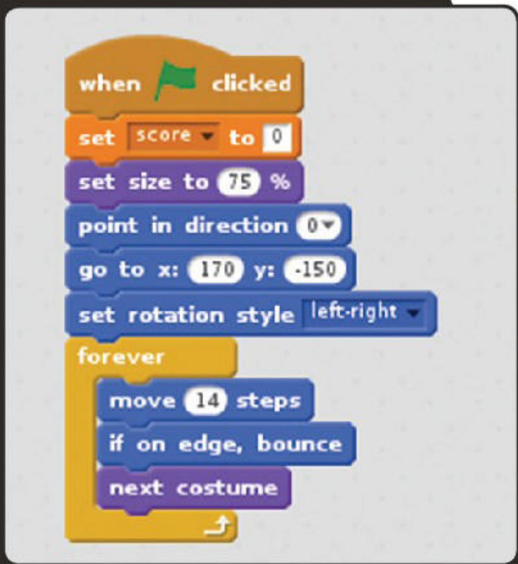
La trasmissione "fire" fa lanciare le banane al gatto (usa il blocco go to mouse-pointer, e scegli Cat1 nel suo menu). Quindi rendi visibili le banane, falle scivolare attraverso lo schermo e nascondile di nuovo.

Fai clic sulla bandiera verde e premi SPAZIO per testare. Il gatto ora lancia le banane.

Top Tip

Fai lo sprite giusto

Accertati di aggiungere gli script allo sprite corretto. Puoi selezionare uno sprite cliccando nella Sprite List.



Top Tip

Blocchi dentro altri blocchi

Alcuni blocchi possono andare uno dentro l'altro. Nel passo 3, il blocco **posizione y** mantiene la posizione y dello sprite mentre la sua posizione x cambia.

E poi?

Siamo grandi fan di Scratch a *The MagPi*, quindi controlla online i nostr numeri arretrati per ulteriori tutorial di Scratch. Il numero 5 include un gioco di memoria, simile a Simple Brian. Il numero 34 ha un quiz a scelta multipla, e il nostro Annuario 2018 include un'introduzione all'elettronica con Scratch. Vedi il numero 76 per una carrellata di risorse per aiutarti a imparare Scratch e non dimenticare che c'è un libro su Scratch nella serie Essentials di *The MagPi* **(magpi.cc/learnscratch)** e il libro *Code Club Book of Scratch* **(magpi.cc/ccbook1)**.

04 Aggiungi la magia della scimmia

Le variabili sono nomi usati per ricordare delle informazioni, ad esempio un punteggio, che potrebbero cambiare. Fare clic sul pulsante Data e poi su Make a Variabile. Dagli il nome 'score', seleziona For All Sprite e fai clic su OK. Aggiungiamo un bersaglio mobile per il gatto per provare a colpirlo. Fai clic sull'icona 'Choose sprite from library' e scegli Monkey2. Seleziona Monkey2 nell'area Sprites e indica lo script detto sopra.

Lo script monkey imposta la nuova variabile score a 0 quando inizia il gioco, quindi muove la scimmia su e giù. Ogni sprite può avere più di un'immagine: il blocco Next Costume le fa ciclare, creando un'animazione.

05 Rilevamento collisioni

La nostra scimmia reagirà quando sarà colpita da un casco di banane. Per farlo, usiamo un blocco if, che controlla se



qualcosa è vero – in questo caso, se la scimmia sta toccando le banane. Se è così, i blocchi al suo interno vengono eseguiti. Ecco, questi blocchi nascondono la scimmia, aggiungono 1 al punteggio e aspettano un secondo prima di mostrare nuovamente la scimmia.

Tutta questa parte di codice va dentro il cilclo forever della scimmia – sotto il blocco denominato next costume – così il programma continua a controllare se la scimmia è stata colpita.

▼ Questi ultimi due script aggiungono un timer e fanno reagire la scimmia alla vista delle banane



06 Tocchi finali

Aggiungiamo un semplice timer per fermare il movimento degli sprite per 30 secondi e fare reagire la scimmia quando vede le banane in arrivo. Aggiungi questi due script per lo sprite monkey. Ora il gioco è completo, perché non provare a sperimentare con esso? Puoi far muovere la scimmia in modo irregolare, anziché scomparire quando viene colpita? Puoi cambiare le posizioni e le direzioni degli sprite per trasformare il gioco, rendendolo più simile a Space Invaders? Che ne pensi di aggiungere più bersagli da colpire? Uno dei migliori modi per imparare a programmare è sperimentare programmi esistenti.

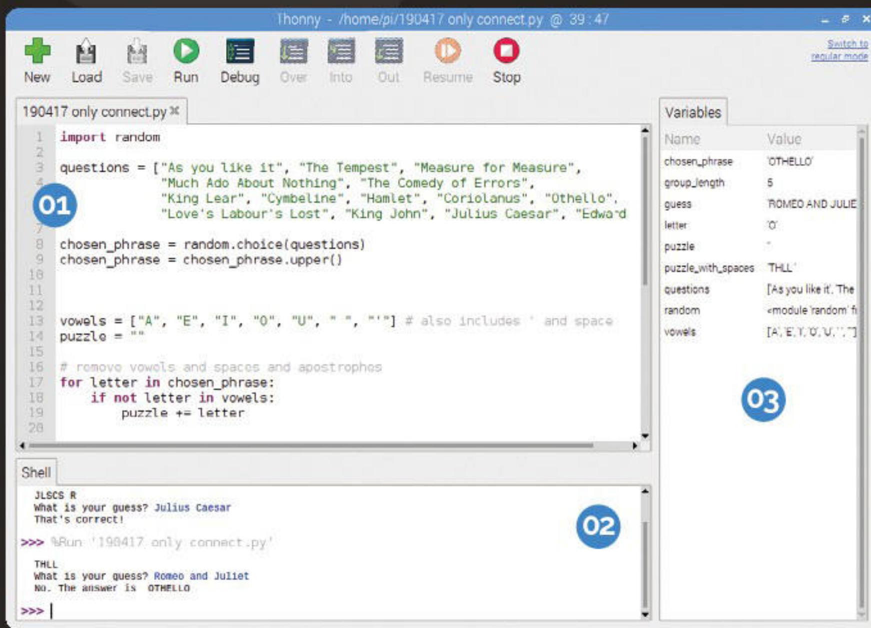
Programmare un gioco a quiz con Python

Realizza il tuo personale gioco a quiz testuale che manipola frasi famose usando il linguaggio Python

Molte persone passano da Scratch a Python, un linguaggio di programmazione che è potente, di facile apprendimento, e molto più semplice da leggere e scrivere rispetto ad altri linguaggi.

Costruiremo un semplice generatore di domande a quiz che toglie le vocali e rimescola gli spazi in una frase. Il giocatore deve capire quale sia la frase.

Useremo Thonny, che fornisce un amichevole ambiente a schermo singolo per far lanciare e testare il codice Python. Come Scratch, l'IDE Thonny (ambiente di sviluppo integrato) è preinstallato in Raspbian con Desktop nel Software Consigliato dal sistema operativo.



01 Digitare e modifica il tuo programma qui

02 Inserisci comandi diretti e vedi qui nella Shell l'input del programma

03 Qui tieni traccia dei dati che il tuo programma sta elaborando

01 Creare una lista di domande

Oltre alle variabili, Python ha le liste, nelle quali può memorizzare delle informazioni multiple. Il nostro programma crea una lista chiamata **`questions`**. Ogni elemento nella lista è un pezzetto di testo, noto come una stringa. In Python, le stringhe sono delimitate da virgolette per mostrare dove iniziano e finiscono. L'intera lista è racchiusa tra parentesi quadre e si usano delle virgole tra gli elementi della lista. Digita il codice qui sotto, salva il programma, quindi clicca su Run. Se funziona, non dovresti avere messaggi di errore nella finestra della Shell.

```
import random

questions = ["As You Like It",
    "The Tempest", "Measure for Measure",
    "Much Ado About Nothing",
    "The Comedy of Errors",
    "King Lear", "Cymbeline",
    "Hamlet", "Coriolanus", "Othello",
    "Love's Labour's Lost",
    "King John", "Julius Caesar",
    "Edward III"]
```

Ottenere l'identazione corretta

Python usa l'identazione per mostrare quali istruzioni appartengono a una funzione, a una dichiarazione if, o a una sezione ripetuta. Come puoi vedere nello step 3 (nella pagina seguente), puoi avere più livelli di indentazione. L'ultima riga appartiene all'istruzione if, e si ripete dentro il ciclo for. Il modo migliore per ottenere l'identazione giusta è ricordarsi i due punti (:) alla fine della riga precedente. Quindi, Thonny aggiungerà la rientranza automaticamente. Se li dimentichi, usa quattro spazi all'inizio della riga, per inserire il rientro. Dovrai comunque correggere quei due punti mancanti!

Cosa Serve

- Raspbian con Desktop e il Software Raccomandato
- Thonny

Top Tip

Fastidiose puntualizzazioni!

Abbi cura di aggiungere i due punti alla fine delle istruzioni **if** e **else**. Il codice non funzionerà senza.

02 Scegli una domanda a caso

Python include dei moduli di codice precompilati che puoi utilizzare, come il modulo random che abbiamo importato nel passo 1. La prima nuova istruzione crea una nuova variabile chiamata **chosen_phrase** e vi inserisce una domanda scelta a caso. La seconda riga converte la stringa contenuta nella variabile in lettere maiuscole. Esegui il programma alcune volte e guarda il valore di **chosen_phrase** nel pannello Variables. Dovresti vedere comparire nomi diversi, anche se questi si possono ripetere.

Aggiungi una linea di spazio tra il codice nel passo 1 e aggiungi il seguente codice:

```
chosen_phrase = random.choice(questions)
chosen_phrase = chosen_phrase.upper()
```

03 Togliere vocali e spazi

Creiamo una nuova lista di caratteri proibiti, principalmente le vocali, ma anche lo spazio e l'apostrofo. L'ultimo elemento della lista **vowels** è un apostrofo all'interno di virgolette. Creiamo una variabile stringa vuota, chiamata **puzzle**. Passando attraverso ogni lettera della frase, controlliamo se questa è nella lista **vowels** e se non è così, la aggiungiamo alla fine della stringa **puzzle**. L'istruzione **for** imposta una ripetizione di un pezzo di codice, chiamato ciclo, o loop. Le istruzioni che sono ripetute sono rientrate da sinistra. Ad ogni ciclo, la variabile **letter** è impostata per contenere il prossimo carattere della stringa **chosen_phrase.**

L'istruzione **if** controlla se la lettera è nella lista **vowels.**

Debugging in Thonny

È possibile eseguire il programma passo – passo per vedere cosa sta facendo, cosa che può aiutarti a trovare gli errori. Fai clic sul pulsante Debug in Thonny, quindi fai clic sul pulsante Over per eseguire le istruzioni una alla volta. Guarda il pannello delle Variabili a destra per vedere come le liste e le stringhe cambiano in ogni fase del programma. Thonny ti aiuta anche a evitare gli errori evidenziando parentesi non chiuse e virgolette doppie.

E poi?

E' possibile trovare codice Python di esempio nella maggior parte dei numeri di *The MagPi*. Il numero 53 (**bit.ly/MagPi53**) include una guida per principianti più approfondita su Python, che tratta le variabili, i cicli con **while** e **for**, ramificazioni con **if**, e le funzioni, che tratteremo a breve in questo numero. Il numero 54 (**bit.ly/MagPi54**) introduce la programmazione orientata agli oggetti, sia in Scratch che in Python. Il numero 73 (**magpi.cc/73**) comprende una panoramica sui libri su Python e sulle risorse online. C'è un libro nella serie Essentials, intitolato *Make Games with Python* (**magpi.cc/gameswithpython**).

Se non lo è, la lettera viene aggiunta alla fine di **puzzle**. Il += significa 'aggiungi alla fine'. Esegui il programma, quindi controlla se lavora correttamente guardando il contenuto di **puzzle** nel pannello Variabili. Non dovrebbe contiene vocali, spazi o apostrofi.

Aggiungi il seguente codice al programma:

```
vowels = ["A", "E", "I", "O", "U", " ", "'"]
puzzle = ""

for letter in chosen_phrase:
    if not letter in vowels:
        puzzle += letter
```

04 Inserire spazi casuali

Ogni carattere della stringa è identificato dal suo numero di posizione, a partire da 0. Il numero si chiama indice e si mette tra parentesi quadre dopo la stringa. Prova questo nella Shell (fai clic sulla riga che inizia con '>>>'). Le istruzioni digitate nella Shell sono eseguite immediatamente. Digita il codice:

```
print("Hello"[1])
```

Ottieni 'e' (perché il primo carattere è il numero 0). Puoi anche ottenerne uno spezzone (chiamato slice) dando un indice di inizio e di fine, come qui:

```
print("Hello"[1:4])
```

Otterrai 'ell' perché l'ultima posizione dell'indice (4) è esclusa. Creeremo una nuova lista, chiamata **puzzle_with_spaces**, aggiungendo blocchi della stringa **puzzle** e uno spazio fino a quando non c'è nessuna stringa **puzzle** a sinistra. Il ciclo **while** ripete le istruzioni indentate fino a che la lunghezza di **puzzle** è maggiore di 0. La variabile **group_length** è un numero casuale (intero) da 1 a 5. Quindi vengono aggiunte molte lettere in **puzzle_with_spaces** dalla parte anteriore di **puzzle**, più uno spazio. Questi caratteri sono quindi ritagliati

> **È facile iniziare con** Python ed è molto più semplice da leggere e scrivere degli altri linguaggi

dalla parte anteriore di **puzzle**. Lo slicing, qui, usa solo un numero, quindi si assume che l'altro sia l'inizio o la fine della stringa.

Aggiungi questo codice al tuo programma:

```
 puzzle_with_spaces = ""

 while len(puzzle) > 0:
     group_length = random.randint(1,5)
     puzzle_with_spaces +=
puzzle[:group_length] + " "
     puzzle = puzzle[group_length:]
```

05 Aggiungere il controllo collisioni

Stampa a schermo **puzzle_with_spaces**. Ora usa la funzione **input()** per chiedere qual è la tua possibile risposta. La risposta va nella variabile **guess**, e viene quindi convertita in maiuscolo per far sì che corrisponda alla risposta corretta, se hai risposto giusto.

L'istruzione **if** verifica se **guess** è la stessa di **chosen_phrase**. Se è così, mostra un messaggio. Altrimenti, vengono eseguite le istruzioni identate sotto l'esecuzione **else**, per dirti la risposta giusta. In Python, un = è usato per mettere un valore in una variabile, ma due (==) sono utilizzati per confrontare gli elementi in un'istruzione **if**.

Aggiungi questo codice alla fine del programma:

```
print(puzzle_with_spaces)
guess = input("What is your guess? ")
guess = guess.upper()

if guess == chosen_phrase:
    print("That's correct!")
else:
    print("No. The answer is ", chosen_phrase)
```

Fai clic sul pulsante Run e dovresti vedere alcune lettere nella Shell "What is your guess?" ("quale la tua ipotesi?") Inserisci una risposta e vedrai "That's correct!" ("È corretto!") oppure "No. The answer is" e la risposta corretta.

Se hai digitato il codice a mano, è potresti visualizzare un messaggio di errore. Controlla il codice riga per riga e confrontarlo con il codice completo in quiz_game.py.

quiz_game.py

Language: **Python**

SCARICA IL CODICE COMPLETO: magpi.cc/github82

```
import random

questions = ["As You Like It", "The Tempest",
"Measure for Measure", "Much Ado About Nothing",
"The Comedy of Errors", "King Lear", "Cymbeline",
"Hamlet", "Coriolanus", "Othello","Love's Labour's Lost",
"King John", "Julius Caesar", "Edward III"]

chosen_phrase = random.choice(questions)
chosen_phrase = chosen_phrase.upper()

vowels = ["A", "E", "I", "O", "U", " ", "'"]
puzzle = ""


for letter in chosen_phrase:
    if not letter in vowels:
        puzzle += letter


puzzle_with_spaces = ""


while len(puzzle) > 0:
    group_length = random.randint(1,5)
    puzzle_with_spaces += puzzle[:group_length] + " "
    puzzle = puzzle[group_length:]


print(puzzle_with_spaces)
guess = input("What is your guess? ")
guess = guess.upper()


if guess == chosen_phrase:
    print("That's correct!")
else:
    print("No. The answer is ", chosen_phrase)
```

Realizza una torcia LED e un gioco elettronico

Scopri quanto può essere facile far lampeggiare le luci e premere i pulsanti usando GPIO Zero e usa le tue nuove abilità per costruire un gioco elettronico

Una delle cose migliori di Raspberry Pi è che puoi facilmente collegare i tuoi personali progetti elettronici. Utilizzando alcuni componenti e la libreria GPIO Zero, puoi programmare un puzzle game dove devi ripetere una sequenza di luci che si allunga ad ogni turno. Probabilmente ricorderai un gioco simile della tua infanzia, ma noi chiamiamo il nostro Simple Brian. Nel numero 77 (**bit.ly/MagPi77**) abbiamo mostrato come usare il codice Python per giocare sullo schermo. In questo numero, ti mostreremo come realizzare il gioco elettronico stesso, basato sulle nuove abilità Python acquisite con le il gioco delle vocali mancanti.

Per prima cosa, ti mostreremo come costruire una torcia accendendo dei LED. Muoviamoci.

01 Collega il tuo primo pulsante

Lo schema del circuito della torcia (Figura 1) mostra una luce a LED collegata ai pin GPIO di un Raspberry Pi tramite una breadboard (vedi magpi.cc/breadboard per un'introduzione all'uso di questa attrezzatura). Spingi il pulsante sulla scheda, quindi utilizza i cavallotti per formare il circuito con il tuo Pi, come indicato dai fili gialli nello schema. Il primo pulsante si collega al pin GPIO 2 da un lato e al binario di massa con l'altro lato. Collegheremo poi quest'ultimo a un pin di massa sul Pi, quindi qualsiasi cosa collegato a quella fila di fori, sarà collegata a massa.

02 Collega il tuo primo LED

Utilizza sempre una resistenza quando colleghi un LED al Pi, per evitare che il LED richieda troppa corrente e danneggi il Pi. Sia il LED che la resistenza si inseriscono direttamente nella breadboard. La corrente scorre dal GPIO 18, attraversa la resistenza, attraversa il LED (accendendolo), fino al binario di massa della breadboard. I LED sono componenti polarizzati: il piedino più corto è il lato negativo, che si collega a massa. Il LED non si accende ancora.

03 Fai una torcia LED

Hai costruito il tuo primo circuito! Testiamolo provando a programmare una torcia. Il codice **torch.py** mostra come utilizzare un LED e un pulsante. Esso importa le parti pertinenti dalla libreria GPIO Zero, quindi imposta un LED chiamato `light`, collegato al pin 18 del GPIO. Il pulsante sul pin 2 è impostato con il nome `button`. Il loop `while True` controlla di continuo se il pulsante è stato premuto. Se è così, la luce viene accesa. Altrimenti, viene tenuta spenta. Prestare attenzione alla capitalizzazione di `LED` e `button` durante l'impostazione.

04 Aggiungi altri pulsanti e LED

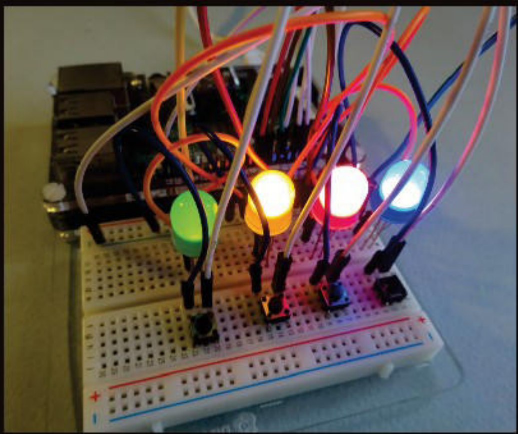
Dai un'occhiata alla Figura 2 (sulla pagina seguente). Sembra complesso all'inizio, ma gli altri tre pulsanti e LED sono collegati allo stesso modo dei primi, semplicemente

Cosa Serve

- 4 LED (idealmente di colori differenti)
- 4 resistenze da 330Ω
- 4 pulsanti Tactile con tasto da 6mm
- breadboard da 400 fori
- 9 cavallotti maschio - femmina
- 8 cavallotti maschio - maschio
- case PiBow con Breadboard Base **pimoroni.com**

E poi?

La documentazione online di GPIO Zero (**magpi.cc/DPyuYc**) fornisce altri esempi di codice, tra cui una fotocamera controllata da un pulsante, un grafico a barre a LED e un sensore di movimento. Noi abbiamo esaminato le risorse utili per l'elettronica di base nel numero 77 e c'è un libro nella serie Essentials chiamata Simple Electronics with GPIO Zero (**magpi.cc/gpio-zero**).



Il gioco finito, con tutte le luci accese per il test

usando pin diversi sul Pi. Tutti i pulsanti (fili gialli) si collegano ai pin del PI della fila interna e i LED (fili blu) ai pin della fila esterna. Abbiamo separato i componenti in questo schema così è più facile vedere come effettuare i collegamenti, ma cerca di allineare i LED e i pulsanti sulla breadboard, in modo che sia più semplice giocare.

05 Provali tutti

Ora puoi testare questi LED e anche i pulsanti. Modifica il codice della torcia per utilizzare `LED(23)` e `Button(3)`, poi esegui il programma per testare se il pulsante lì vicino accende il LED. poi testa `LED(24)` e il `Button(4)`, infine `LED(25)` e `Button(17)`. I pulsanti dovrebbero essere accanto al LED che illuminano.

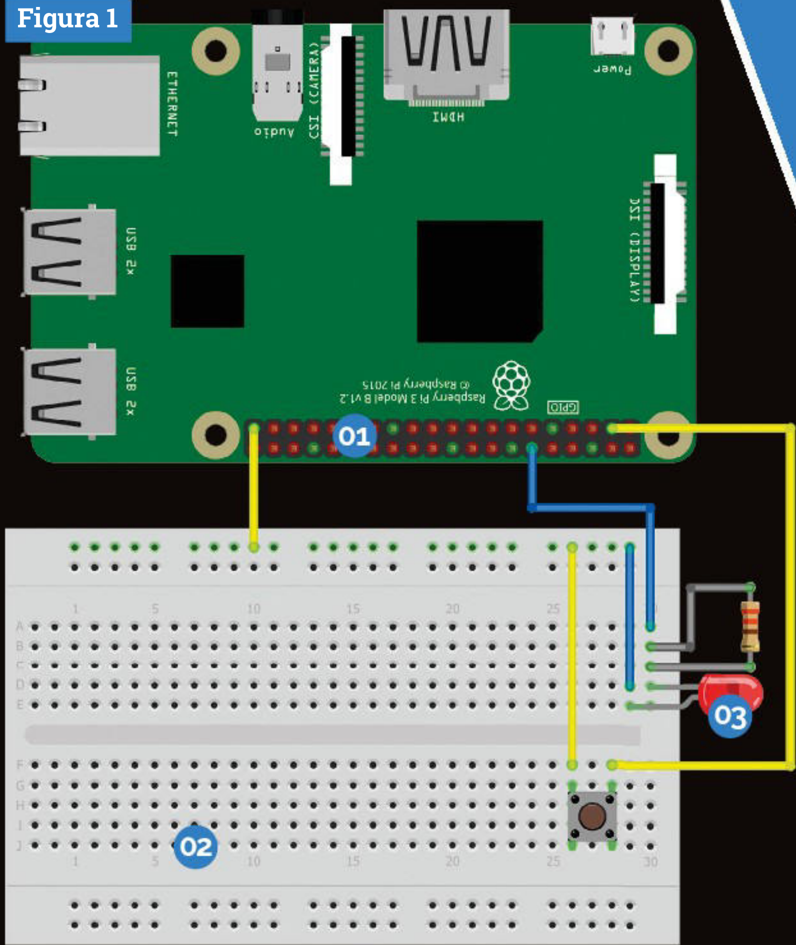
06 Costruisci un gioco elettronico

Ora siamo pronti per iniziare a realizzare il semplice gioco Brian (vedi **brian.py**). Inizia con la configurazione di una lista di pulsanti e una lista dei relativi LED associati. Crea anche una lista vuota chiamata `sequence`, che useremo per la sequenza di luci che il giocatore deve ripetere. Ad ogni turno, questa si allungherà.

07 Aggiungere funzioni

Le funzioni ti consentono di raggruppare un insieme di istruzioni per poterle riutilizzare. Per poter usare una funzione, la devi prima definire. Per definire una funzione, si usa `def`, seguita dal nome della funzione, () e due punti. Le parentesi servono a contenere qualsiasi informazione che tu voglia inviare alla funzione, ma se non sono necessarie, le parentesi sono vuote. Per identificare quali istruzioni appartengono a una funzione, queste vengono indentate (rientrate rispetto alle altre). Le funzioni `lights_on()` e `lights_off()` utilizzano un ciclo per

Figura 1



01 Conta i pin per vedere dove connetterti con i fili

02 La breadboard semplifica l'inserimento dei componenti e crea rapidamente i circuiti

03 Usando questo semplice circuito (con LED e pulsante) è possibile creare una torcia a pulsante

torch.py

Linguaggio: **Python**

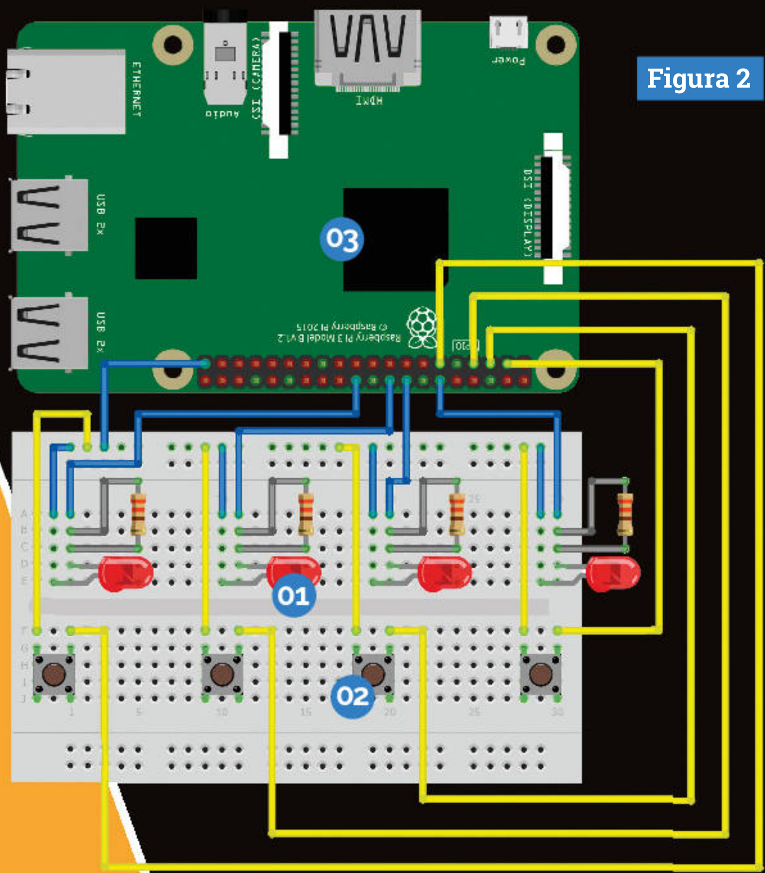
SCARICA IL CODICE COMPLETO
magpi.cc/github82

```
# Torch demo
from gpiozero import Button, LED

light = LED(18)
button = Button(2)

while True:
    if button.is_pressed:
        light.on()
    else:
        light.off()
```

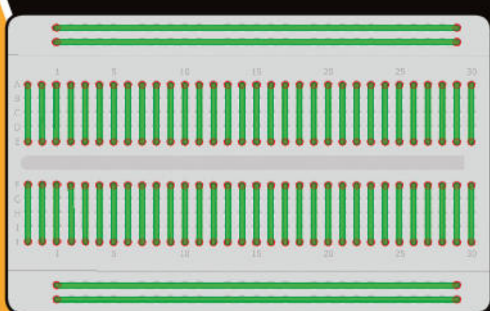
Figura 2



01 Proprio come il famoso omonimo, il nostro gioco Simple Brian presenta quattro luci a LED che lampeggiano in sequenza da memorizzare

02 Il circuito contiene anche quattro pulsanti. Bisogna premerli per ripetere la sequenza di luci che abbiamo memorizzato

03 Il cuore del nostro progetto è il Raspberry Pi. I pulsanti e le luci sono collegati al nostro computer con fili. Il codice nel computer fa lampeggiare le luci e tiene d'occhio i pulsanti che stiamo premendo



▲ Su una breadboard, i binari lungo i bordi sono collegati in una lunga fila, e nelle brevi linee di punti nel centro, questi sono collegati tra loro

scorrere tutti gli elementi della lista `leds`, mettendoli in `led`, alternando led acceso e spento. La funzione `flash_all()` mostra come ripetere il ciclo un determinato numero di volte, in questo caso 3. Il loop accende e spegne le luci, con una pausa di 0,25 secondi dopo ogni modifica usando `sleep(0.25)`.

08 Test luci lampeggianti

Dopo aver inserito le funzioni (cioè dopo la riga 23), è possibile testare il programma aggiungendo `flash_all()` come ultima riga e quindi eseguirlo. Tutte le luci dovrebbero lampeggiare insieme, tre volte. Cancella questa riga di test prima di continuare. nella riga 25, il programma esegue la funzione `lights_off()` per essere certi che e luci siano spente prima dell'inizio del gioco.

09 Aggiungere le sequenze

Ora entriamo nel ciclo principale del gioco, sotto `while True` (riga 27). Tutto da qui in poi è indentato per mostrare che appartiene a quel ciclo, ripetuto indefinitamente. La sequenza di gioco `sequence` inizia come lista vuota, quindi la prima cosa che facciamo è aggiungere un LED. Scegliamo un LED a caso usando `random.choice()` e lo aggiungiamo alla fine della lista `sequence` usando il metodo `append()`. Un metodo è una funzione integrata in Python che è possibile applicare a una lista. Sono disponibili altri metodi per inserire e rimuovere degli elementi, e per ordinare le liste, tra le altre cose.

10 Riprodurre le sequenze della lista

Le luci lampeggiano tre volte usando la funzione `flash_all()` prima che inizi la sequenza, per mostrare che questo è l'inizio della sequenza. Poi viene utilizzato un loop che prende ogni LED dalla lista `sequence` e lo mette in `light` a sua volta. Accende il LED, fa una breve pausa, quindi lo spegne. Un'altra breve pausa quindi ovviamente ci possono essere più lampeggi della stessa luce se si ripete nella sequenza. Nel primo round, c'è solo una luce nell'elenco della sequenza, ma man mano che il gioco procede, questo ciclo si allungherà. A questo punto, puoi eseguire il programma per vedere la sequenza di luci allungarsi gradualmente, senza che il giocatore abbia la possibilità di indovinare.

11 Ottenere la risposta del giocatore

Ottenuta l'ipotesi di risposta del giocatore usa un ciclo simile a quello che riproduce la sequenza di

brian.py

SCARICA IL CODICE COMPLETO
magpi.cc/github82

Linguaggio: **Python**

```
001. from gpiozero import Button, LED
002. from time import sleep
003. import random
004.
005. buttons = [Button(2), Button(3), Button(4),
     Button(17)]
006. leds = [LED(18), LED(23), LED(24), LED(25)]
007.
008. sequence = []
009.
010. def lights_on():
011.     for led in leds:
012.         led.on()
013.
014. def lights_off():
015.     for led in leds:
016.         led.off()
017.
018. def flash_all():
019.     for _ in range(3):
020.         lights_on()
021.         sleep(0.25)
022.         lights_off()
023.         sleep(0.25)
024.
025. lights_off()
026.
027. while True:
028. # Aggiungi una nuova luce alla fine della sequenza
029.     new_light = random.choice(leds)
030.     sequence.append(new_light)
031.
032. # lampeggia tutti prima
033.     flash_all()
034.
035. # fai la sequenza
036.     for light in sequence:
037.         light.on()
038.         sleep(0.5)
039.         light.off()
040.         sleep(0.25)
041.
042. # prendi l'input dall'utente
043.     for light in sequence:
044.         guess = None
045.         while guess == None:
046.             for button in buttons:
047.                 if button.is_pressed:
048.                     # converti il pulsante nel
numero della lista
049.                     guess = buttons.index(button)
050.
051.         if leds[guess] == light:
052.             light.on()
053.             sleep(0.5)
054.             light.off()
055.             sleep(0.25)
056.         else:
057.             print("Hai perso al livello ",
len(sequence))
058.             for _ in range(10):
059.                 light.on()
060.                 sleep(0.15)
061.                 light.off()
062.                 sleep(0.15)
063.             sequence = []
064.             break
```

accensione delle luci. Si fa strada attraverso la lista **sequence**, accetta l'ipotesi di risposta e controlla se corrisponde all'elemento corrente nella sequenza. Ci sono tre loop uno dentro l'altro. Il programma imposta la variabile **guess** su **None**, un valore speciale in Python. Poi un ciclo **while** continua a ripetersi fino a quando la variabile **guess** cambia. Al suo interno, un altro loop scorre la lista **buttons**, controllandone ognuno a turno per vedere se è stato premuto. Se è così, la variabile **guess** è cambiata, termina così il ciclo **while.** Il programma converte il pulsante premuto dal giocatore nel numero di indice nella lista e lo inserisce nella variabile **guess**. In questo modo, possiamo abbinare il pulsante al suo LED, che sarà nella stessa posizione nella lista dei LED. Puoi trovare la posizione di un elemento in una lista usando **listname.index(item)**.

12 Controllo risposta del giocatore

Siamo ancora dentro il ciclo che scorre la sequenza, come mostra il rientro della linea 51. Ora controlliamo se la luce che il giocatore ha fornito (**leds[guess]**) corrisponde alla luce corrente nella sequenza. Se è così, la luce viene accesa e poi di nuovo spenta. Se le due luci non corrispondono, il giocatore ho fatto un errore. Possiamo dire quanto è lunga una lista usando **len(listname)**. Usiamo la lunghezza della sequenza per indicare al giocatore a quale livello è arrivato. La luce corretta viene poi fatta lampeggiare rapidamente dieci volte. L'elenco delle sequenze viene svuotato per iniziare un nuovo gioco, e l'istruzione **break** interrompe il ciclo **for** quando si è ottenuto l'input del giocatore. Quando il giocatore ha o indovinato tutte le luci, o le ha sbagliate, il gioco si ripete dalla riga 28, aggiungendo una nuova luce alla sequenza.

Top Tip

Ripetizione underscore

La riga `for _ in range(3)` ripete il codice indentato tre volte. Il _ mostra che non c'è bisogno di usare il numero di ciclo. Spesso, invece, useresti un nome di variabile.

PROGETTI DI MODELLLISMO FERROVIARIO

Vuoi potenziare il tuo hobby per i trenini? Fallo con il Pi!

Vediamo così tanti hobbisti usare Raspberry Pi in modi incredibili e nuovi. In molti casi, questo rende un hobby più accessibile (come nella robotica), o compiti precedentemente difficili un po' più semplici (come negli indossabili avanzati).

Uno dei più antichi hobby elettronici in circolazione, il modellismo ferroviario, è costituito da intricati binari e servizi ferroviari splendidamente miniaturizzati, a lungo coltivato nei garage e scantinati. Come ogni altra cosa, la tecnologia in questo campo è migliorata negli anni e molte persone hanno trovato il modo di incorporare un Raspberry Pi nei loro plastici ferroviari. Quindi, saltiamo a bordo e vediamo quali progetti possiamo trovare!

STAZIONI

Scegli la tua destinazione

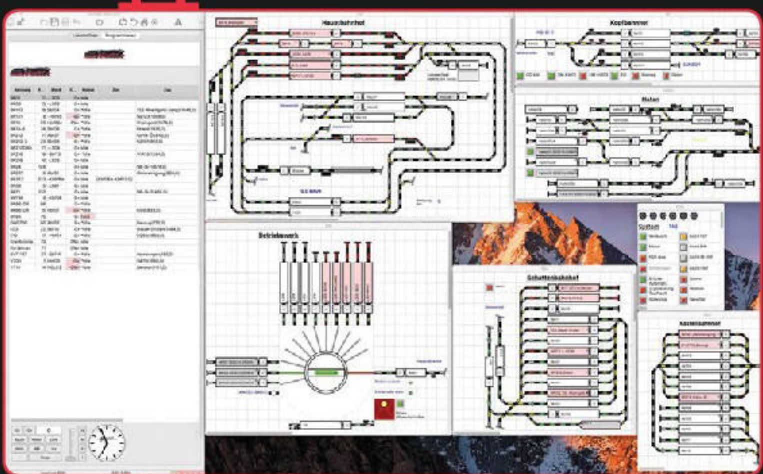
SOFTWARE DI CONTROLLO

Alcuni ottimi modi per iniziare con il modellismo ferroviario gestito dal Pi

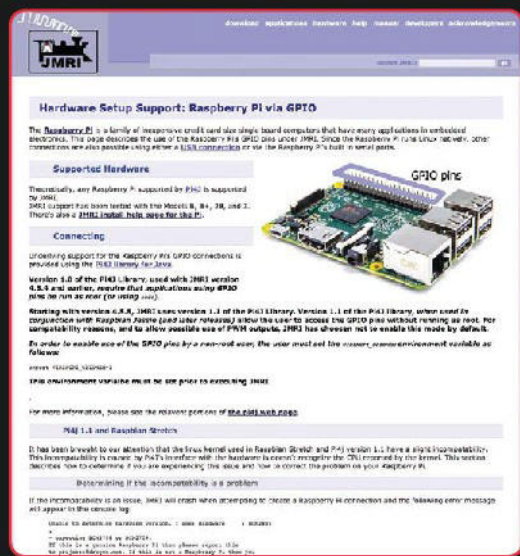
Uno dei principali vantaggi di aggiungere un Raspberry Pi a una ferrovia in miniatura è avere la possibilità di realizzare del codice personalizzato che possa essere eseguito esattamente come lo vuoi tu. Ci sono alcuni software che ti possono aiutare in questo...

Rocrail | magpi.cc/WaTTaF

Rocrail è un software gratuito multipiattaforma per il controllo dei trenini. Permette all'utente di controllare, completamente o parzialmente, un intero sistema ferroviario tramite automazione e controllo manuale, a seconda delle sue preferenze, e include più visuali, e una interfaccia basata su blocchi per la configurazione dell'automazione. Ha anche un client web, che rende il controllo manuale è più facilmente accessibile da qualsiasi luogo. Questo controllo include auto, accessori e illuminazione
- È piuttosto potente!



Con Rocrail si può accedere a diverse interfacce utente da qualsiasi dispositivo connesso al Web



JMRI (Java Model Railway Interface) può girare su Raspberry Pi e puoi usarlo per controllare la tua ferrovia

JMR-Pi | magpi.cc/puVuFE

Questa è una porting per Raspberry Pi di JMRI, un software di controllo basato su Java, adatto al modellismo ferroviario. È descritto semplicemente come un modo di ascoltare e inviare messaggi sul 'command bus'. In combinazione con il software DecoderPro (**magpi.cc/wRpmek**), può essere usato per controllare un plastico ferroviario.

JMR-Pi su Raspberry Pi funzionerà come una control box / server headless a cui connettersi da remoto - e grazie alle sue dimensioni, si nasconde molto meglio in un plastico ferroviario rispetto a un computer tradizionale.

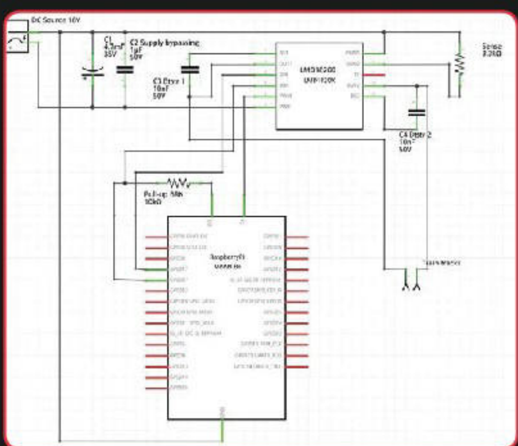
STAZIONI DI COMANDO

Hardware per controllare le ferrovie

> "Rocrail ha persino un client web, che rende il controllo manuale facilmente accessibile da ovunque"

dccpi | magpi.cc/ZFMZBJ

Una libreria Python appositamente ottimizzata per ilRaspberry Pi, ti consente di lavorare con il protocollo DCC, che viene utilizzato nel controllo dei modellini ferroviari. È una libreria discretamente semplice, in grado di controllare direzione e velocità, così come alcune luci, tramite il GPIO. Richiede anche un po' di alimentazione extra per assicurarsi che fornisca un segnale ai binari, ma è una soluzione divertente, molto fai-da-te per controllare un treno. Utilizza la libreria WiringPi, che tecnicamente potrebbe anche essere utilizzata per controllare la ferrovia, ma dccpi è molto più avanzato.



▲ Raspberry Pi ha bisogno di un circuito supplementare, come questo, per fornire effettivamente il segnale ai binari



SPROG
sprog-dcc.co.uk

Un HAT per iRaspberry Pi che lo trasforma in una Interfaccia DCC.

Digitrax decoder
digitrax.com

Digitrax realizza un numero di decoder che lavorano sulla sua LocoNet.





Hornby Elite
magpi.cc/mLFCkg

Un po' caro, ma è compatibile con un sacco di questi software.

Verifica l'hardware supportato in qualsiasi software decidi di utilizzare!

FANTASTICI PROGETTI DI MODELLISMO FERROVIARIO

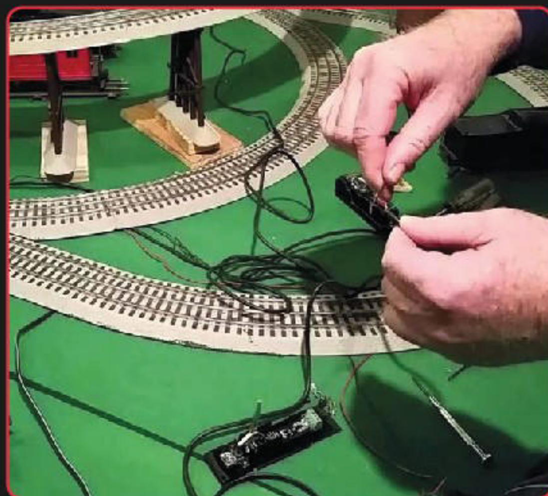
Ecco qui alcune cose incredibili fatte con un plastico ferroviario e un Raspberry Pi!

Controllo interruttore treni Lionel

magpi.cc/yiLZhx

I nostri amici della Dexter Industries hanno messo assieme questa guida qualche tempo fa, mostrandoti come controllare degli interruttori in un plastico ferroviario usando una miscela di Arduino e Raspberry Pi. Questo usa un metodo più diretto rispetto a DCC, come se se tu fisicamente mettessi e togliessi alimentazione o operassi manualmente sugli interruttori.

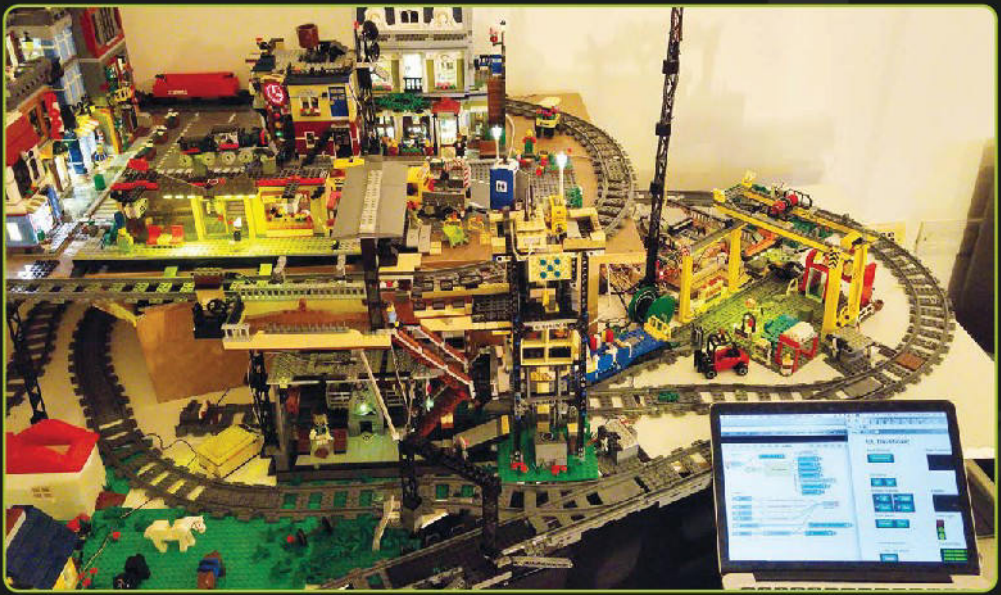
Sebbene questo metodo sia abbastanza diverso dagli altri, è un modo interessante per programmare l'installazione, e ti consente di utilizzare i sensori collegati al Raspberry Pi un po' più facilmente per implementare dell'automazione o aggiungere i tuoi pulsanti manuali o controlli. Forse potresti stampare in 3D il tuo controller DCC con questo.

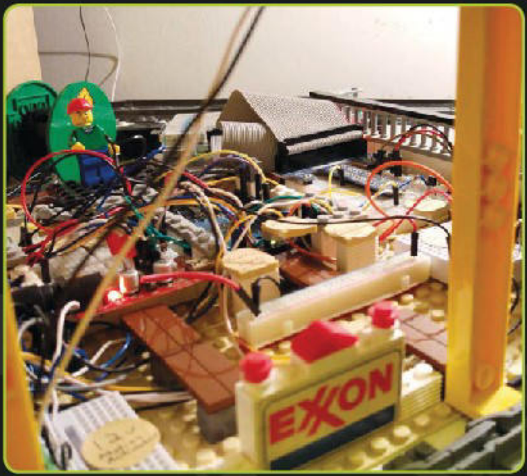




▲ Ti servirà un relè per per gestire 20 V AC, quindi assicurati di procurarti il relè corretto e togli alimentazione prima di collegare tutto

> Internet of Lego è un progetto vivente in cui ho iniziato a imparate tutto sull'Internet degli oggetti





▲ Diversi dispositivi Pi e Arduino funzionano insieme per controllare la città

Internet of Lego

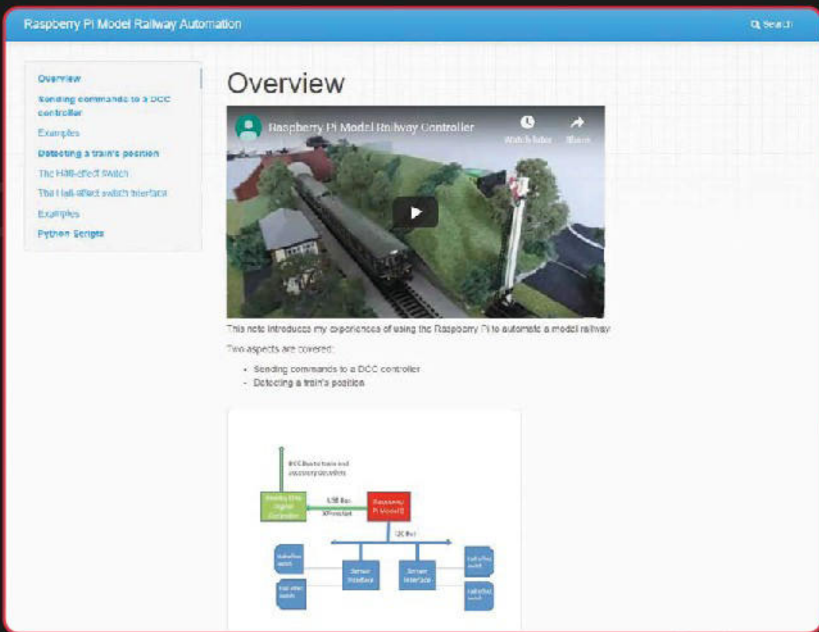
internetoflego.com

Più che un modello di ferrovia, questo è una intera città di Lego, inclusa la ferrovia, che è controllata da una miscela di Raspberry Pi e Arduino. Abbiamo parlato con il suo creatore, Cory Guynn, nel numero 48 di The MagPi (**magpi.cc/48**).

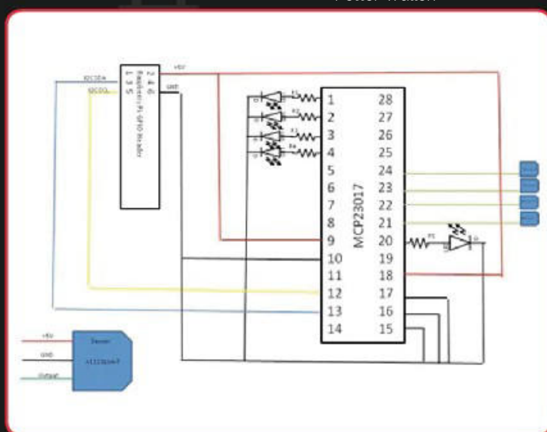
"Internet of Lego è un progetto vivente in cui ho iniziato a imparare tutto sull'internet degli oggetti", ci ha detto Cory allora. Il sistema ferroviario è la sua parte preferita, siccome ama vedere le cose in movimento. C'è un sistema di pianificazione dei treni costruito sulle API del Transport for London, mostrando la destinazione su uno schermo OLED e cambiando il percorso per abbinarlo.

Utilizza sensori WiFi, infrarossi, cablati e altro ancora, per controllare non solo il treno e gli interruttori, ma anche gli schermi e le altre parti della gigantesca città IoT Lego.

> It's a great way to learn about all the low-level electronics and code involved in a pre-existing system



▼ Scopri tutto sugli interruttori a effetto Hall dalla configurazione di Petter Wallen

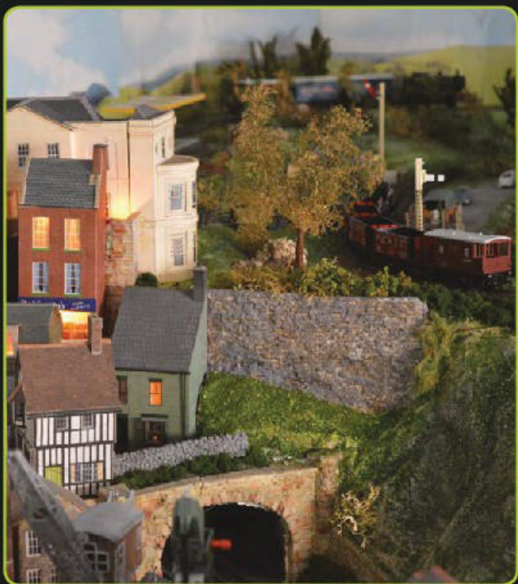


Automazione Modelli Ferroviari con Raspberry Pi

magpi.cc/FXtwjt

Se vuoi entrare davvero nel succo della questione del controllo del to plastico ferroviario, ti consigliamo di dare un'occhiata al progetto di Peter Wallen. Esso fa uso del protocollo XPressNet, che è uno dei tanti modi in cui puoi controllare i dispositivi DCC. Sebbene il codice sia scritto in Python, invia comandi alla riga di comando per attivare specifici indirizzi di oggetti collegati al Pi, quindi è molto a basso livello.

È un ottimo modo per conoscere a basso livello tutta elettronica e il codice coinvolti su un sistema preesistente, però, sono disponibili alcuni script di esempio per aiutarti a iniziare a integrarlo nella tua personale ferrovia.



Le luci si accendono e si spengono a sezioni a seconda del ciclo giorno-notte

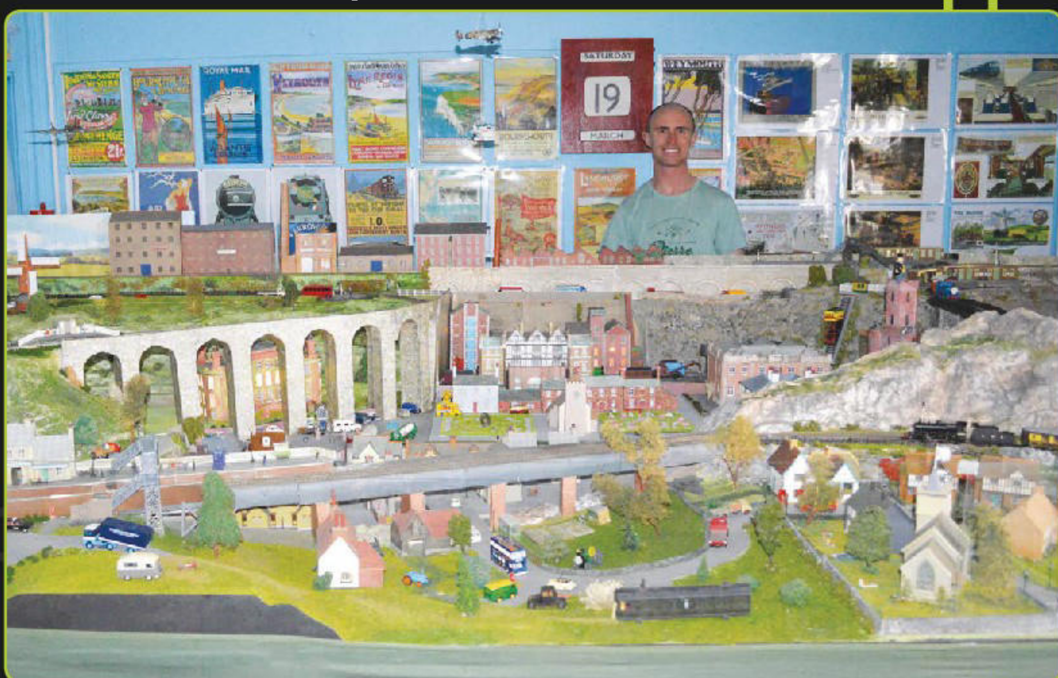
Plastico ferroviario di Wimborne

magpi.cc/wScJqw

Il modellismo ferroviario è molto più di far girare una locomotiva sul suo binario - c'è anche la riproduzione in miniatura del paesaggio attorno ad esso! Wimborne Model Town nel Dorset è la sede si una grande ferrovia in miniatura in una città in miniatura molto intricata. Tutte le case hanno le luci, e ci sono anche diverse lampade e lampioni nel plastico.

Alcuni anni fa, un volontario di nome Terry Coles ha aggiornato il sistema di illuminazione alla città in miniatura, in modo che potesse essere controllato tramite un Raspberry Pi. Ci sono due gruppi di luci: quelle in miniatura, e una grande striscia di LED che funge da sole per il modellismo ferroviario. Esiste infatti un ciclo giorno-notte nel modello ferroviario, con i LED che illuminano tutto come se fosse mezzogiorno, prima di spegnersi lentamente a riprodurre il tramonto. Come questo succede, le luci nella riproduzione in miniatura iniziano a accendersi in sezioni, trasformando il tutto in una notte buia costellata dalle piccole luci del modello.

È programmato in C e utilizza WiringPi. Una delle le parti più intelligenti del progetto è aver caricato il sistema operativo in memoria in modo che se c'è una interruzione di alimentazione, non si corra il rischio che si possa corrompere la scheda SD.



Imparare l'intelligenza artificiale con Raspberry Pi

Con queste risorse, il tuo Pi può pensare da solo. **PJ Evans** spiega l'apprendimento (automatico)

Google AI Education

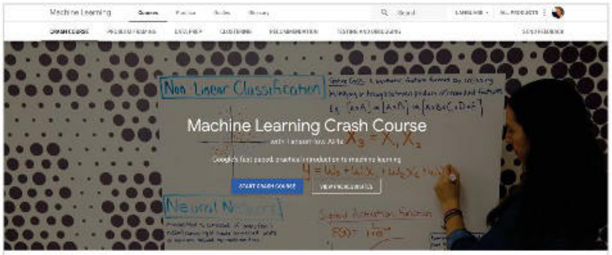
AUTORE **Google**

Prezzo: Gratuito

ai.google/education

Il Raspberry Pi è un potente strumento quando si tratta di intelligenza artificiale (IA) e apprendimento automatico (machine Learning -ML). La sua capacità di elaborazione, abbinata a un fattore di forma ridotto e al basso consumo, lo rendono un'ottima scelta per la robotica intelligente e progetti embedded. Google è campione nel posizionare il Pi nel mondo IA, il suo sistema di riconoscimento vocale AIY, dato in omaggio con questa stessa rivista (numero 57, non più disponibile in forma cartacea - **bit.ly/MagPi57**).

Il sito AI Education di Google è un posto ideale per iniziare il tuo viaggio sul machine learning. Se vuoi davvero capire come funzionano IA e ML 'sotto il cofano', ci sono molti principi da comprendere prima ancora di arrivare alla programmazione. Google fornisce una suite autoguidata che inizia con un 'Crash Course' sul machine learning, quindi si espande per coprire le basi di inquadratura del problema e di raccolta dati. Il principale corso online comprende 25 lezioni per oltre 15 ore (circa, puoi impostare il tuo ritmo) e si presenta sotto forma di risorse di lettura, sezioni interattive, esercizi di programmazione e video tutorial. Il tutto è poi supportato da una sostanziale raccolta di corsi di approfondimento. Una risorsa superba.



Una guida di autoapprendimento per aspiranti professionisti dell'apprendimento automatico

Siti indispensabili

Fornitori di popolari strumenti di IA/ML

GOOGLE CORAL

Recentemente trattata sul numero 79 di *The MagPi*, questo nuovo entusiasmante acceleratore USB di Google trasforma le capacità di intelligenza artificiale di Raspberry Pi aggiungendo un processore dedicato alle reti neurali. Inoltre, non perderti il sito AIY di Google.
coral.withgoogle.com

OPENCV

Strumento scelto da molti costruttori di robot, la libreria Open Source Computer Vision non solo dona la vista Pi, ma anche la capacità di 'comprendere' ciò che vede. Uno strumento potente per il riconoscimento intelligente degli oggetti.
opencv.org

TENSORFLOW

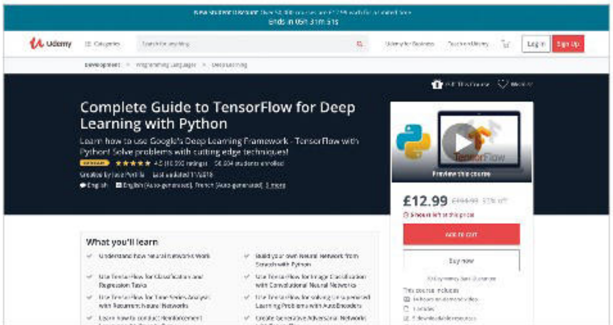
Se vuoi implementare il machine learning su un Pi, è probabile che per farlo utilizzerai TensorFlow. Il sito ufficiale non solo offre una documentazione completa, ma anche una serie di corsi.
tensorflow.org



Complete Guide to TensorFlow for Deep Learning with Python

AUTORE
Jose Portilla

Prezzo:
149,99€ (spesso scontato)
magpi.cc/rbAdNE



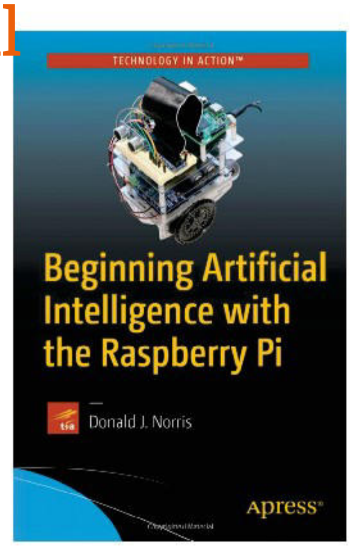
TensorFlow è, senza dubbio, la più popolare libreria software per il machine learning su Raspberry Pi. Se sei desideroso di iniziare e scrivere il codice, TensorFlow sarà probabilmente lo strumento da te scelto. Puoi trovare un'ottima presentazione di TensorFlow sul numero 71 di *The MagPi* (download gratuito: **bit.ly/MagPi71**), ma se cerchi una immersione profonda, Udemy offre un completo video corso di 14 ore che copre non solo la teoria del machine learning, ma anche la pratica della realizzazione del software con esempi reali e esercitazioni di programmazione. Se stai cercando un approccio pratico all'apprendimento, potrebbe essere perfetto. Non storcere il naso davanti al prezzo alto – questo corso è spesso in promozione e veniva 9,99€ al momento della traduzione di questa pagina.

Beginning Artificial Intelligence with the Raspberry Pi

AUTORE
Donald J Morris

Prezzo:
28€ (in digitale 20€)
magpi.cc/VfrZbO

Questo libro è perfettamente su misura per la Community Raspberry Pi. Non solo illustra i principi dietro concetti come reti neurali, logica fuzzy e superficialmente anche il deep learning, ma fornisce anche pratici e divertenti progetti da costruire e programmare. A partire da semplici esempi di apprendimento, puoi far giocare il tuo Pi a tris e Nim. Lungo il percorso, i progetti sono resi divertenti attraverso l'uso del connettore GPIO del Pi, utilizzando LED e interruttori per dare vita al codice. Farai quindi progressi nella robotica, esplorando l'evitamento degli ostacoli e la ricerca della luce. Una curva di apprendimento costante



culmina nella costruzione di 'Alfie', il tuo personale veicolo robot con intelligenza artificiale. Se fantastichi di realizzare il vincitore delle prossime Pi Wars, questa potrebbe essere la lettura perfetta.

Imparare dagli esempi

Impari mentre fai? Prova questi

COME FARE PROGETTI DI IA FAI DA TE USANDO GOOGLE TENSORFLOW E RASPBERRY PI

Una raccolta di progetti IA/ML da costruire o da cui prendere ispirazione. Dalle introduzioni a TensorFlow a una vasta gamma di progetti inclusi il magic mirror e una strabiliante macchina per ordinare i cocomeri!
magpi.cc/gLsMHK

IA SU RASPBERRY PI CON INTEL NEURAL COMPUTE STICK

Come Google, anche Intel ha rilasciato una coprocessore USB basato su rete neurale. Questo tutorial è un'ottima "guida per iniziare", che ti parla dell'installazione e della tua prima app per il riconoscimento facciale.
magpi.cc/jSShvE

RASPBERRY PI POKÉDEX

PyImageSearch è una risorsa incredibile per imparare OpenCV. Questo dettagliato il tutorial è l'ideale per le menti più giovani, usa un Pi e il touch-screen ufficiale per creare un Pokédex che può 'riconoscere' peluche Pokémon.
magpi.cc/Ergqjd

